

**REMARKS**

Reconsideration of the rejections set forth in the Office Action is respectfully requested.

Rejection under 35 USC 103

Claims 37-42 and 45-52 were rejected under 35 USC 103 as unpatentable over Rothschild (U.S. Patent Application Publication No. 2002/0016718) in view of Primak (U.S. Patent No. 6,389,448), and Martin (U.S. Patent No. 6,263,368). Claims 43-44 and 51-52 were rejected under 35 USC 103 as unpatentable over Rothschild in view of Primak and Martin, and further in view of Carr (U.S. Patent No. 6,301,617) and Liu (U.S. Patent No. 5,031,089). These rejections are respectfully traversed in view of the following arguments.

In the response to arguments section, the Examiner stated in two places (Page 12, lines 8-12 and Page 12, lines 18-20) that “one cannot show non-obviousness by attacking references individually where the rejections are based on combinations of references.” Applicants do not disagree with this general statement of the law of obviousness.

However, applicants present a hypothetical: Assume that a claim has 4 elements (elements A, B, C, and D). In the hypothetical, the claim is rejected over a combination of references (reference #1 and reference #2). If the applicant is able to show that reference #1 does not show element D, and that reference #2 also does not show element D, then it naturally follows that the combination of references does not show element D. Attacking the references individually to explain how the references work to show that neither element shows element D is one way of showing that the combination only shows elements A, B, and C, not A, B, C, and D as claimed.

In this situation, applicants argued that none of the references teaches or suggests that the complexity of the task to be performed should be used in connection with selecting a server to service a task. Since the claims are directed at this feature, and none of the references teach or suggest this feature, it naturally follows that the combination of references does not teach or suggest this feature.

In the Response to arguments section, Office Action at page 12, lines 14-18, the Examiner has taken the position that Primak teaches assessing the complexity of a task to be performed in connection with load balancing, citing Primak at Col. 4, lines 30-37 and Fig. 2b. Specifically, the Examiner stated that Primak “discloses monitoring CPU capacity, CPU load,

number of tasks being performed and the number of connections” which the Examiner stated “has the same meaning as complexity in the specification of the present application as shown in paragraph 22”.

Fig. 2b of Primak shows a router 30 connecting three servers 10(a), 10(b) and 10(c) to the Internet. Each server has a Load Balancing module 12. A client passes a connection request (SYN packet) to the router which multicasts the connection request to all of the servers (Primak at Col. 3, lines 46-48: “After receiving the SYN packet from the client computer 60, the router 30 multicasts the SYN packet to all of the servers 10(a)-(c).”).

Primak does not teach or suggest that the complexity of the SYN packet or the task associated with the SYN packet should be used in making a load balancing decision when deciding which of the servers will process the SYN packet. Paragraph 22 of the instant specification, cited by the Examiner in the Office Action, specifies that “the level of complexity of a task will be based on the amount of resources that are required to execute the task, such as processor time or memory, or the complexity may be a function of the time that is required to execute the command.” These features look at the new task that is to be allocated, not the existing tasks that are already being processed by the various processors. Since the “task” in Primak is the SYN packet associated with establishment of a new connection, to find a corollary between paragraph 22 and Primak the Examiner would need to show that Primak looks at the complexity of the SYN packet. Primak does not do this as discussed below.

In Primak, the load balancing module 12 of each server evaluates whether its associated server will handle the connection request by (1) calculating a pseudo-random number; and (2) determining the relative availability of each server. (Primak at Col. 3, lines 49-54: “The evaluation process involves calculating a pseudo-random number for each SYN packet and determining the relative availability of each server.”) The load balancing module either passes the SYN packet (connection request) onto the server’s stack or discards it based on its evaluation of the SYN packet. (Primak at Col. 3, lines 54-57).

At Col. 3, line 57 to Col. 4, line 4, Primak explains how the pseudo-random number is generated and, at Col. 4, lines 4-6, Primak states that each Load Balancing module generates an identical random number.

Starting at Col. 4, line 8, Primak explains how the relative availability of a server is determined. Specifically, Primak states that “The relative availability of a server is a function of

its overall capacity and current load.” (Primak at Col. 4, lines 8-9). Primak then explains that the load balancing module determines the relative available capacity of the server by looking at the CPU capacity, CPU load, number of current processes or tasks being performed, and the number of existing connections. (Primak at Col. 4, lines 12-22). All of these things that are monitored are associated with the server CPU. None of the mentioned conditions are associated with the complexity of the new task (SYN packet).

In the portion of the reference cited by the Examiner (Primak at Col. 4, lines 30-37), Primak teaches that an agent 14 on each server will periodically transmit the availability of the server to the other servers so that each of the load balancing agents knows how busy each of the other servers is. For convenience, Col. 4, lines 28-37 of Primak is reproduced below:

In an embodiment of the present invention as shown in FIG. 4a, the agent 14 records the availability information of its associated server in a pre-defined memory location 16. The agent 14 can periodically transmit the availability information of its associated server either to other servers in the cluster (FIG. 2a) or to a coordinating device 20 (FIG. 2b). As shown in FIG. 2a, each agent 14 broadcasts its respective availability information to all of the servers in the cluster. Alternatively, as shown in FIG. 2b, each agent 14 transmits its respective availability information only to the coordinating device 20. It is appreciated that the coordinating device 20 may be a server or a general purpose computer.

As is clear from reading this portion of Primak in context, the load balancing modules on each server are attempting to make decisions in parallel as to which of the servers is going to service a particular connection request. To prevent servers from making different decisions, the load balancing modules on each of the servers are all required to use the same information when making the load balancing decision for a particular SYN packet. Since the load balancing modules make this determination based on a pseudo-random number and the relative availability of each server, (Primak at Col. 3, lines 52-54), each load balancing module is required to generate identical pseudo-random numbers (Primak at Col. 4, lines 4-6), and have identical relative availability information (Primak at Col. 4, lines 30-37). By providing each load balancing module 12 with a copy of the SYN packet, ensuring that each load balancing module will generate the same pseudo-random number and make a decision based on the same relative availability information, Primak is able to provide a system where distributed load balancing decisions are made by each of the servers.

Independent claim 37 recites that the method includes the step of receiving medical image data with embedded instructions associated with a task. Claim 37 further recites that the network service extracts the instructions, determines a level of complexity of the task to be performed, and selects one of the image archive resources based in part on the “level of complexity of the task to be performed”.

The portion of Primak cited by the Examiner does not teach or suggest that the server should be selected based on a complexity of the task to be performed. Rather, each server in Primak bases its selection on (1) a pseudo-random number and (2) the relative availability of each server. Nothing in this looks at the complexity of the task to be performed. Specifically, nothing is looking at the complexity of the connection request associated with the SYN packet.

Contrary to the Examiner’s assertion in the Office Action on page 12, lines 13-18, applicants respectfully submit that Primak fails to teach or suggest a load balancing process that selects an image archive resource based, at least in part, on the “level of complexity of the task to be performed.

In connection with rejecting claims 37-42 and 45-52, the Examiner cited three references: Rothschild, Primak, and Martin. In connection with rejecting claims 43-44 and 51-52, the Examiner further cited Carr and Liu. However, none of these cited references teach or suggest that a load balancing process should select an image archive resource based, at least in part, on the “level of complexity of the task to be performed. To put this in the context of the hypothetical set forth above, all of the cited references fail to teach or suggest part D of the claim. Accordingly, the combination of Rothschild, Primak, and Martin alone or further in view of Carr and Liu, fails to render the claims obvious.

Rothschild teaches a medical image management system that may be implemented by an Application Service Provider to provide network based delivery and storage of medical images. The medical image management system allows users to have access to the medical images securely over the network and provides special clinical and visualization applications centrally for the remote users. (Rothschild at paragraphs 136-140). Rothschild does not teach or suggest a system that intercepts medical images as they are transported across the network and then load-balances the images to image archive resources based on the complexity of the task required to be implemented on the medical images and the available capacity of the various networked image archive resources.

Applicants explained above that Primak, like Rothschild, fails to teach suggest that the complexity of a task should be factored into the load balancing determination.

Martin teaches that network server link loading may be monitored and used as the basis of performing load balancing between servers (Martin at Col. 3, 36-42). Martin does not, however, teach or suggest that the load balancer should look at the complexity of the task when selecting a server.

Carr teaches a way of virtualizing servers by using a virtual uniform resource locator. When a request is received for the virtual uniform resource locator, one of the servers is selected based on the status of the servers. The actual uniform resource locator of the selected server is then returned. (Carr at abstract; Col. 2, lines 31-47). Carr does not teach or suggest that a task associated with the request, or that the complexity of the task, should be used in connection with selecting a server to service the request. Rather, Carr is focused on a particular way that a group of servers may be addressed (using a virtual URL) and how the URL may be changed once one of the group of servers is selected.

Liu was cited by the Examiner as teaching a priority level of a task; the Examiner did not assert that Liu teaches or suggests evaluating a complexity of the task when performing load balancing.

Thus, none of the applied references teaches or suggests that the complexity of the task to be performed should be used in connection with selecting a server to service the task. Accordingly, the combination of the applied references likewise fails to teach or suggest this feature. In view of this explanation, applicants respectfully request that the rejections of the pending claims under 35 USC 103 be withdrawn.

#### Conclusion

Applicants respectfully submit that the pending claims are patentable over the cited combination of references. If the Examiner believes that further amendments are required to overcome the cited references, the Examiner is respectfully requested to call the undersigned to discuss this application. Likewise, if the Examiner has any questions about the cited art or the remarks presented herein, the Examiner is requested to contact the undersigned to discuss this application.

Serial No. 10/805,993

Extension of Time

Applicants request a two month extension of time to respond to the Office Action. The fee for the extension of time is being submitted herewith. If any additional fees are due in connection with this filing, the Commissioner is hereby authorized to charge payment of the fees associated with this communication or credit any overpayment to Deposit Account No. 501602 (Ref: 16410BAUS01U).

Respectfully Submitted

Dated: February 26, 2010

/John C. Gorecki/  
John C. Gorecki  
Registration No. 38,471

Anderson Gorecki & Manaras LLP  
P.O. Box 553  
Carlisle, MA 01741  
Tel: (978) 264-4001  
Fax: (978) 264-9119  
[john@gorecki.us](mailto:john@gorecki.us)